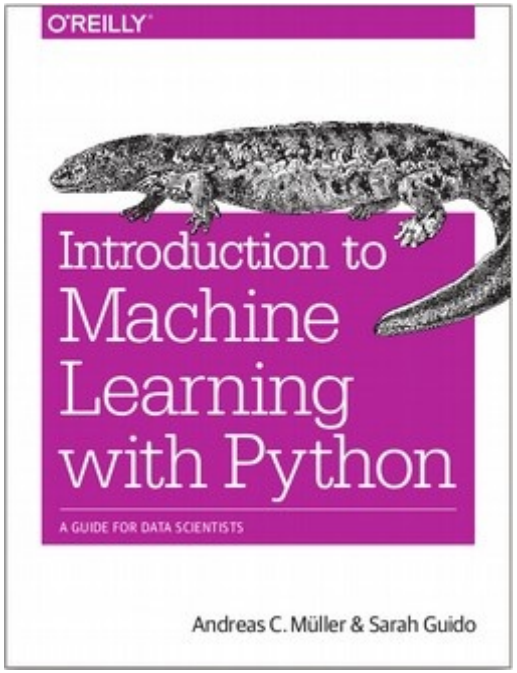
Open Source for (Data) Science

Andreas Müller

Associate Research Scientist
Columbia University
Scikit-learn Technical Committee

universitätbonn ais

Amazon Machine Learning



NYU Center for Data Science

COLUMBIA UNIVERSITY
Data Science Institute

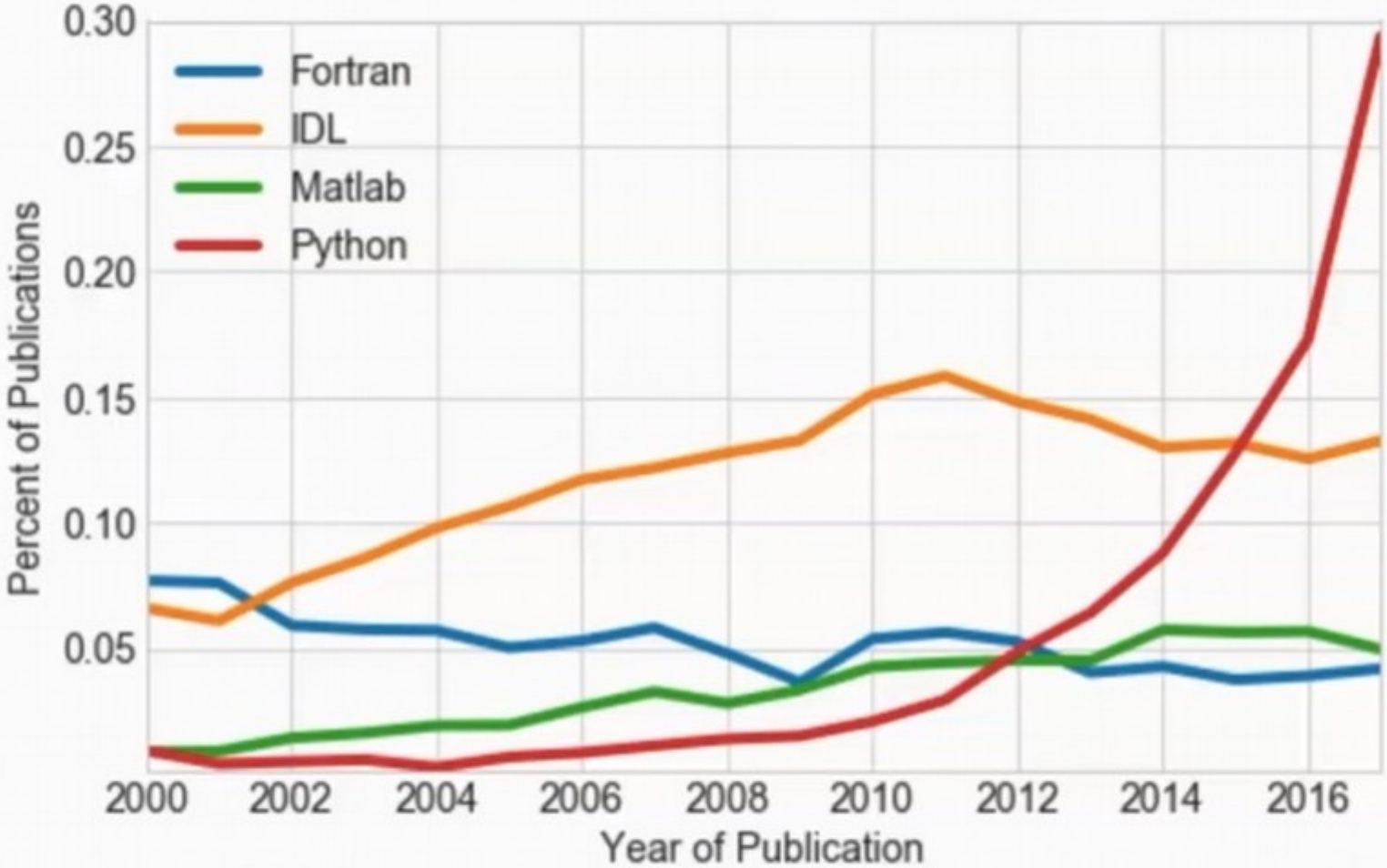
Why Software Matters

“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship.

The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”

- Buckheit and Donoho (1995)

Mentions of Software in Astronomy Publications:



Thanks to Juan Nunez-Iglesias, Thomas P. Robitaille, and Chris Beaumont.

Compiled from NASA ADS (code).

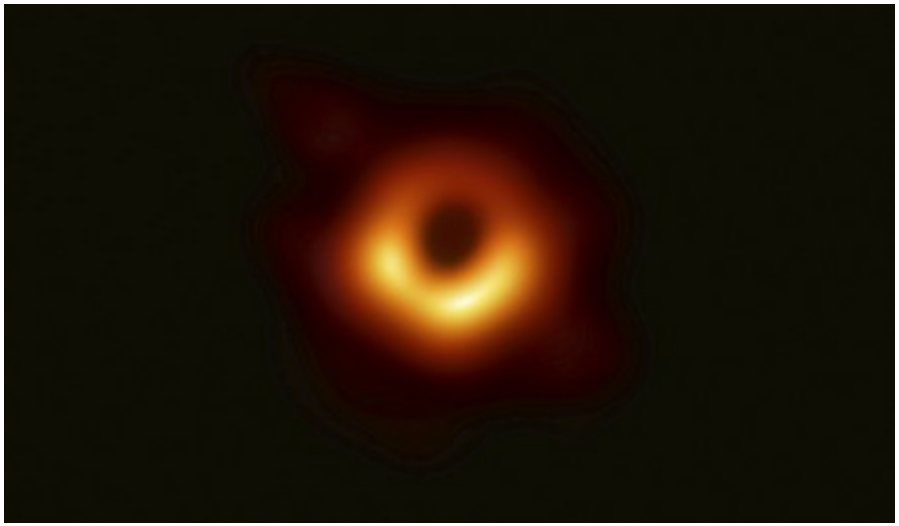
Impact

“The scientific Python ecosystem is critical infrastructure for the research done at LIGO.”

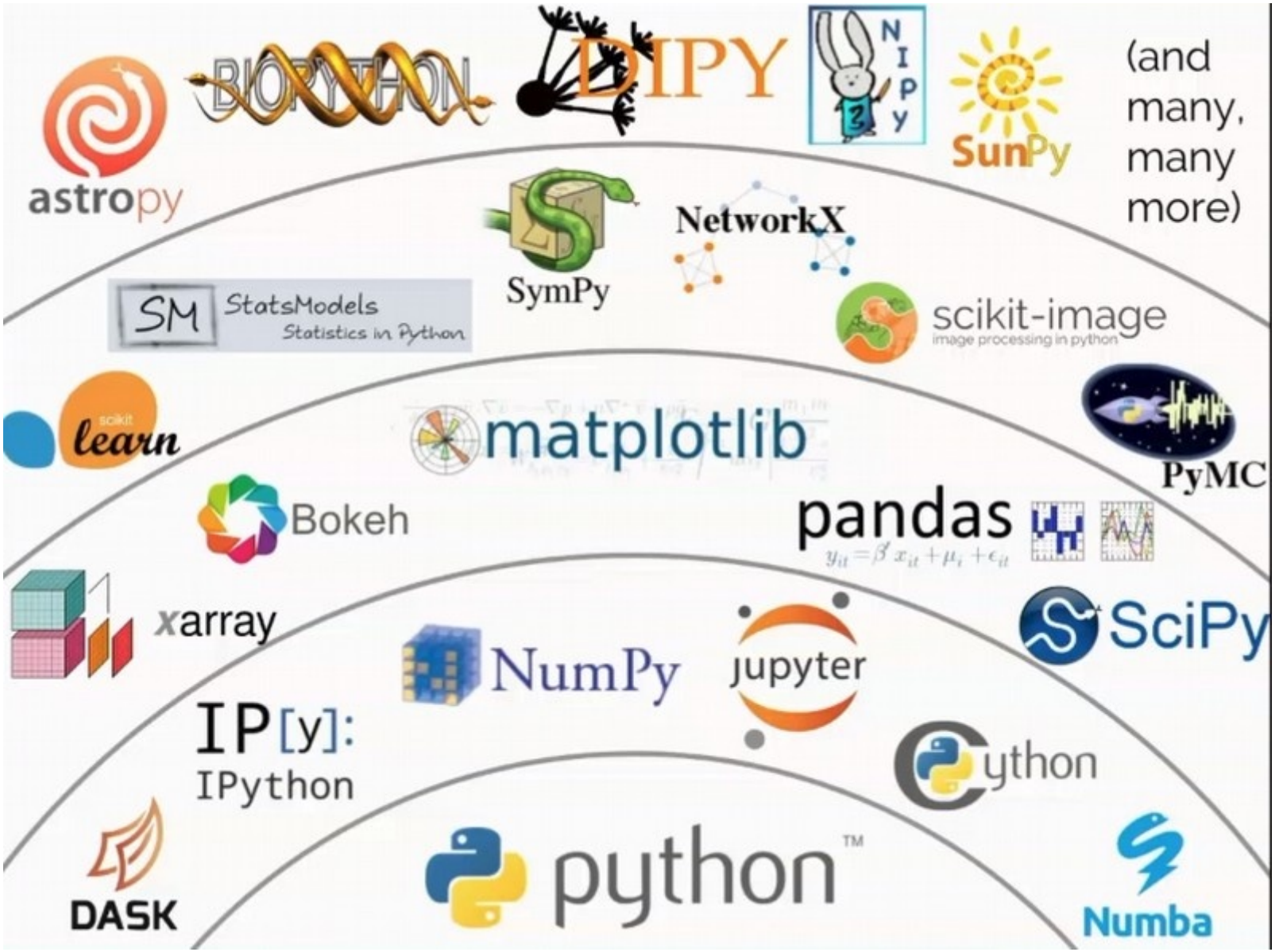
David Shoemaker, LIGO Scientific Collaboration

These [SciPy/PyData] are fundamental to training biologists to have the computational skills they need to be effective in today's research realm.

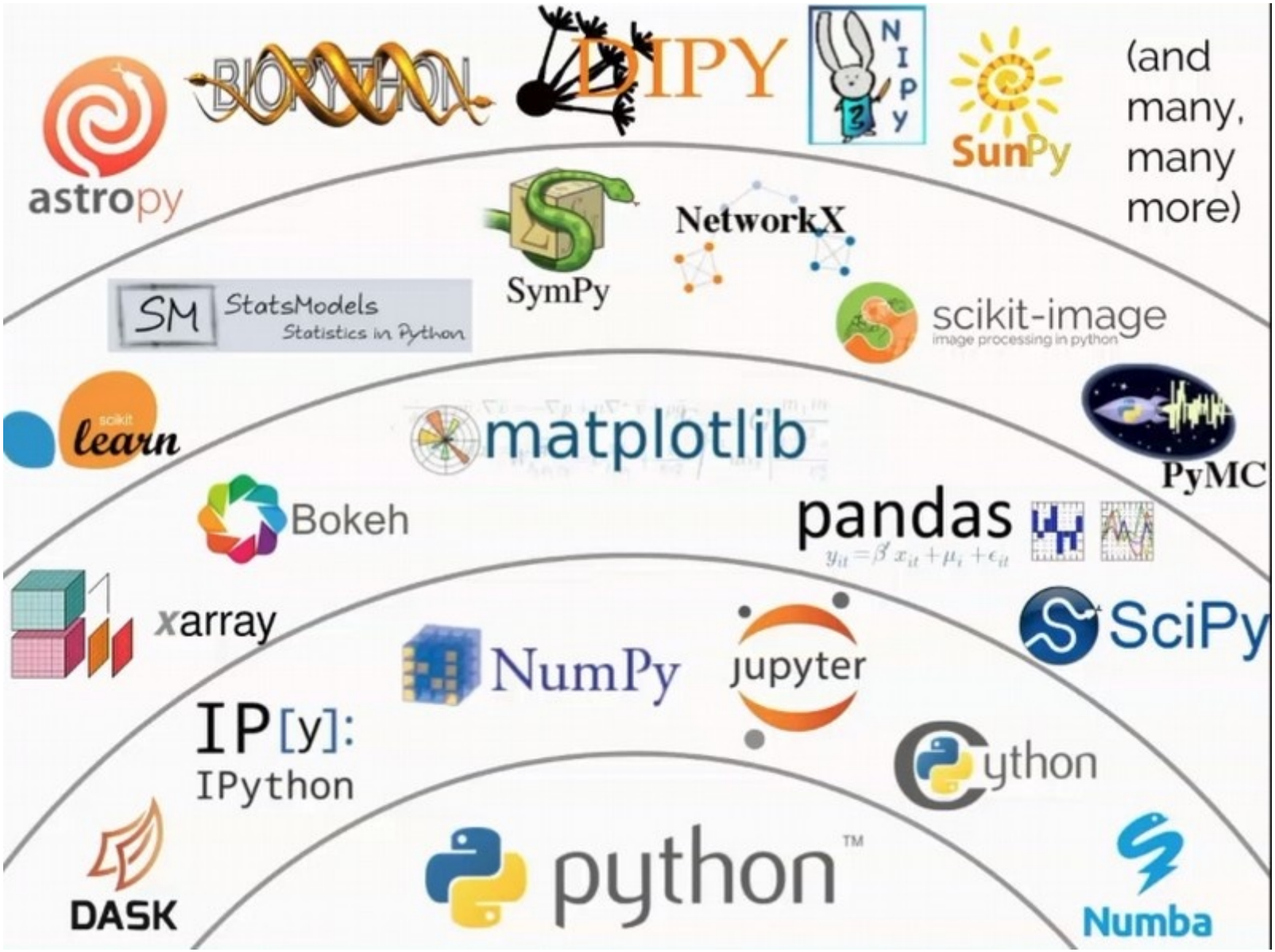
Bonnie Hurwitz, Hurwitz lab, iPlant



The Scientific Python Ecosystem



The Scientific Python Ecosystem



Transforming Science Through Cyberinfrastructure

NSF's Blueprint for a National Cyberinfrastructure Ecosystem for Science and Engineering in the 21st Century

Executive Summary

Twenty-first century science and engineering (S&E) research is being transformed by the increasing availability and scales of computation and data. The national cyberinfrastructure (CI) ecosystem has thus become a key catalyst for discovery and innovation and now plays a critical role in ensuring US leadership in S&E, economic competitiveness and national security, consistent with NSF's mission. The vision and blueprint presented in this document have been developed by the NSF Office of Advanced Cyberinfrastructure (OAC) on behalf of NSF based on a synthesis of multiple community inputs through advisory bodies, requests for information (RFIs), workshops and conferences, and national initiatives.

A new vision. NSF envisions ***an agile, integrated, robust, trustworthy and sustainable CI ecosystem that drives new thinking and transformative discoveries in all areas of S&E research and education.*** This vision embodies the following overarching principles:

Transforming Science Through Cyberinfrastructure

NSF's Blueprint for a National Cyberinfrastructure Ecosystem for Science and Engineering in the 21st Century

No mention of
- Python
- SciPy
- R
- Open Source

Executive Summary

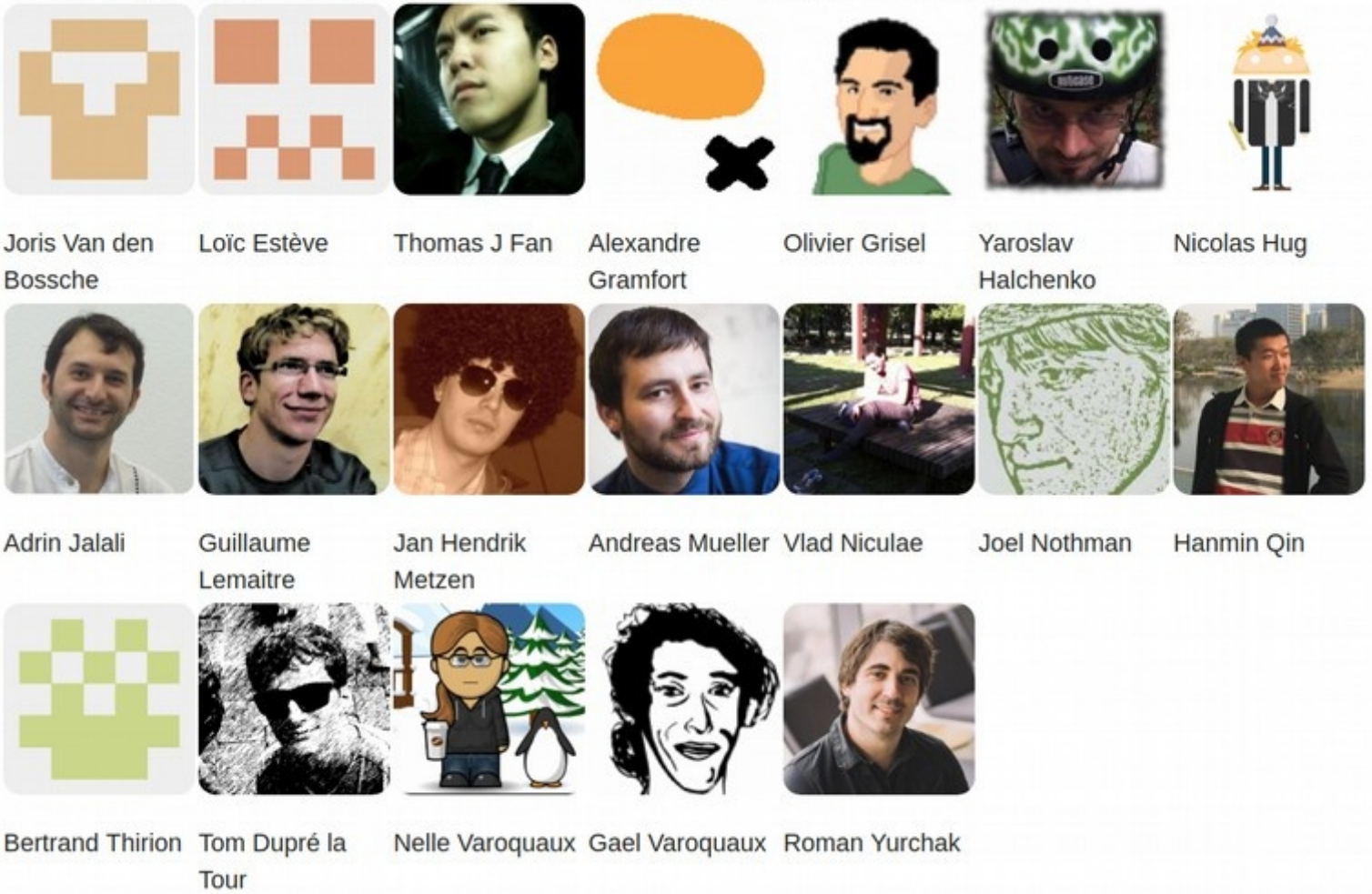
Twenty-first century science and engineering (S&E) research is being transformed by the increasing availability and scale of computation and data. The national cyberinfrastructure (CI) ecosystem has thus become a key catalyst for discovery and innovation and now plays a critical role in ensuring US leadership in S&E, economic competitiveness and national security, consistent with NSF's mission. The vision and blueprint presented in this document have been developed by the NSF Office of Advanced Cyberinfrastructure (OAC) on behalf of NSF based on a synthesis of multiple community inputs through advisory bodies, requests for information (RFIs), workshops and conferences, and national initiatives.

A new vision. NSF envisions ***an agile, integrated, robust, trustworthy and sustainable CI ecosystem that drives new thinking and transformative discoveries in all areas of S&E research and education.*** This vision embodies the following overarching principles:

Machine Learning with Python and Scikit-learn

Authors

The following people are currently core contributors to scikit-learn's development and maintenance:



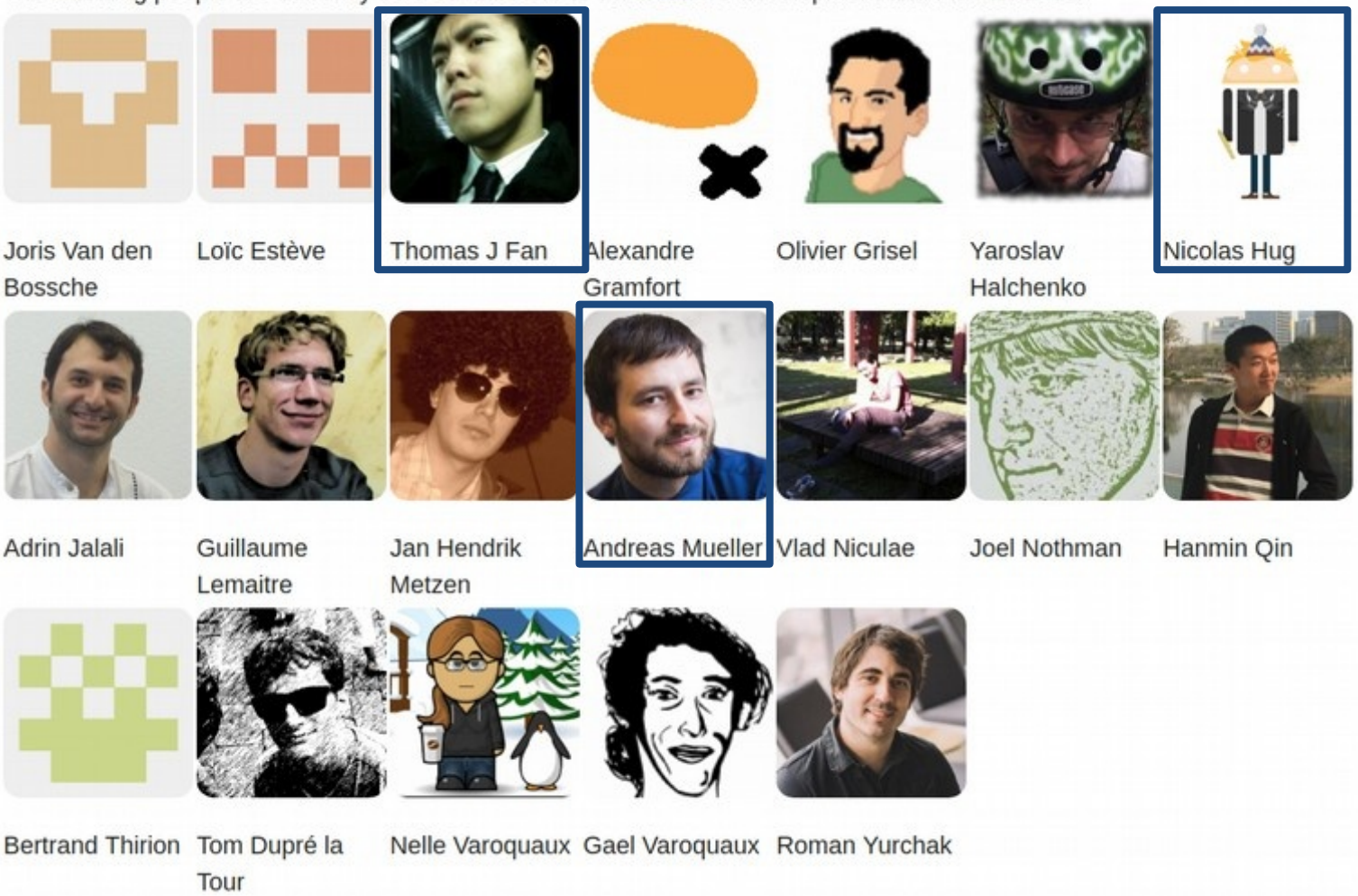
1,323 contributors

Emeritus Core Developers

Alexander Fabisch
Alexandre Passos
Angel Soler Gollonet
Arnaud Joly
Chris Gorgolewski
David Cournapeau
David Warde-Farley
Eduard Duchesnay
Fabian Pedragosa
Gilles Louppe
Jacob Schreiber
Jake Vanderplas
Jaques Grobler
Jarrod Millman
Kyle Kastner
Lars Buitinck
Manoj Kumar
Mathieu Blondel
Matthieu Brucher
Noel Dawe
Paolo Losi
Peter Prettenhofer
Raghav Rajagopalan
Robert Layton
Ron Weiss
Satrajit Ghosh
Shiqiao Du
Thouis (Ray) Jones
Vincent Dubourg
Vincent Michel
Virgile Fritsch
Wei Li

Authors

The following people are currently core contributors to scikit-learn's development and maintenance:



1,323 contributors

Emeritus Core Developers

Alexander Fabisch
Alexandre Passos
Angel Soler Gollonet
Arnaud Joly
Chris Gorgolewski
David Cournapeau
David Warde-Farley
Eduard Duchesnay
Fabian Pedragosa
Gilles Louppe
Jacob Schreiber
Jake Vanderplas
Jaques Grobler
Jarrod Millman
Kyle Kastner
Lars Buitinck
Manoj Kumar
Mathieu Blondel
Matthieu Brucher
Noel Dawe
Paolo Losi
Peter Prettenhofer
Raghav Rajagopalan
Robert Layton
Ron Weiss
Satrajit Ghosh
Shiqiao Du
Thouis (Ray) Jones
Vincent Dubourg
Vincent Michel
Virgile Fritsch
Wei Li

Our Mission:

Commoditize and Democratize Machine Learning

Scikit-learn: Machine learning in Python 17173 2011

F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, ...

Journal of machine learning research 12 (Oct), 2825-2830

About 100 citations / day

About 4 citations since this meeting started!

Scikit-learn: Machine learning in Python 17173 2011

F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, ...

Journal of machine learning research 12 (Oct), 2825-2830

About 100 citations / day

About 4 citations since this meeting started!

scikit-learn / **scikit-learn** ✓ Used by 58,454

Scikit-learn: Machine learning in Python 17173 2011
F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, ...
Journal of machine learning research 12 (Oct), 2825-2830

About 100 citations / day
About 4 citations since this meeting started!

scikit-learn / **scikit-learn** ✓ Used by 58,454

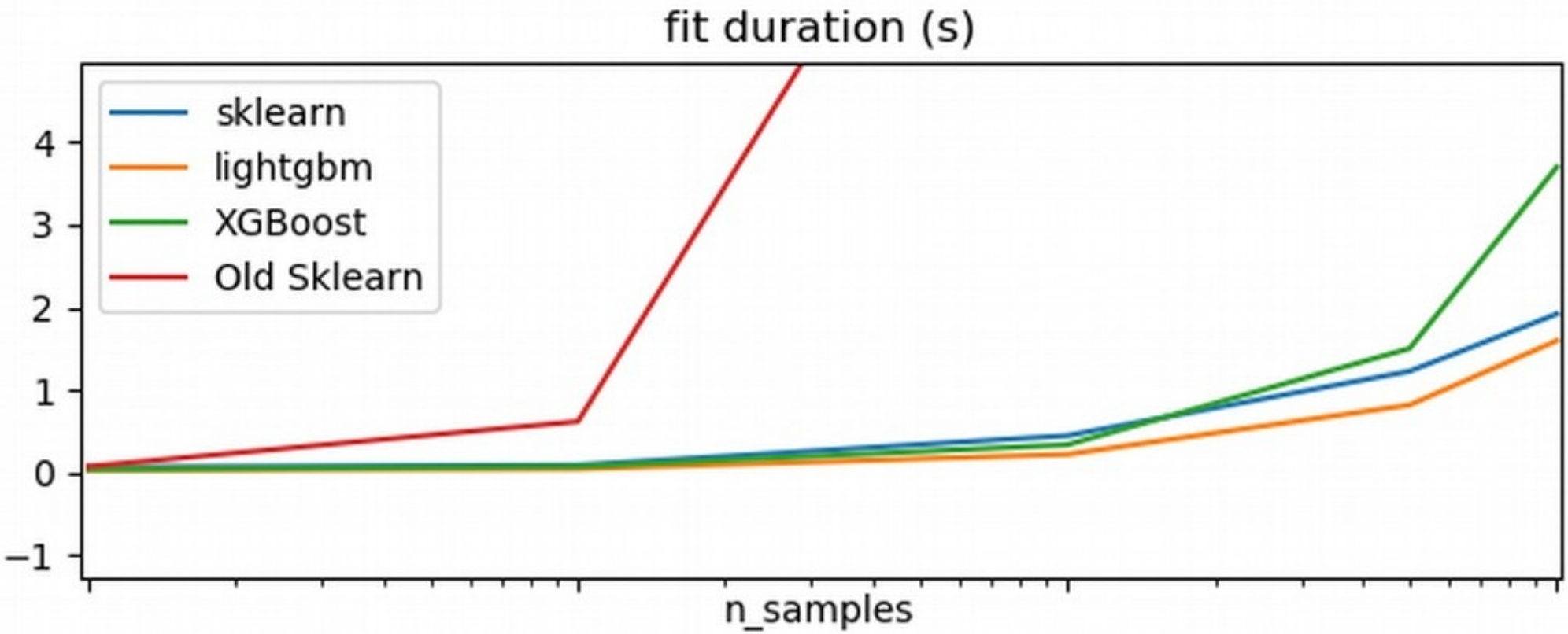
François Chollet
@fchollet
Following

Replying to @ylecun

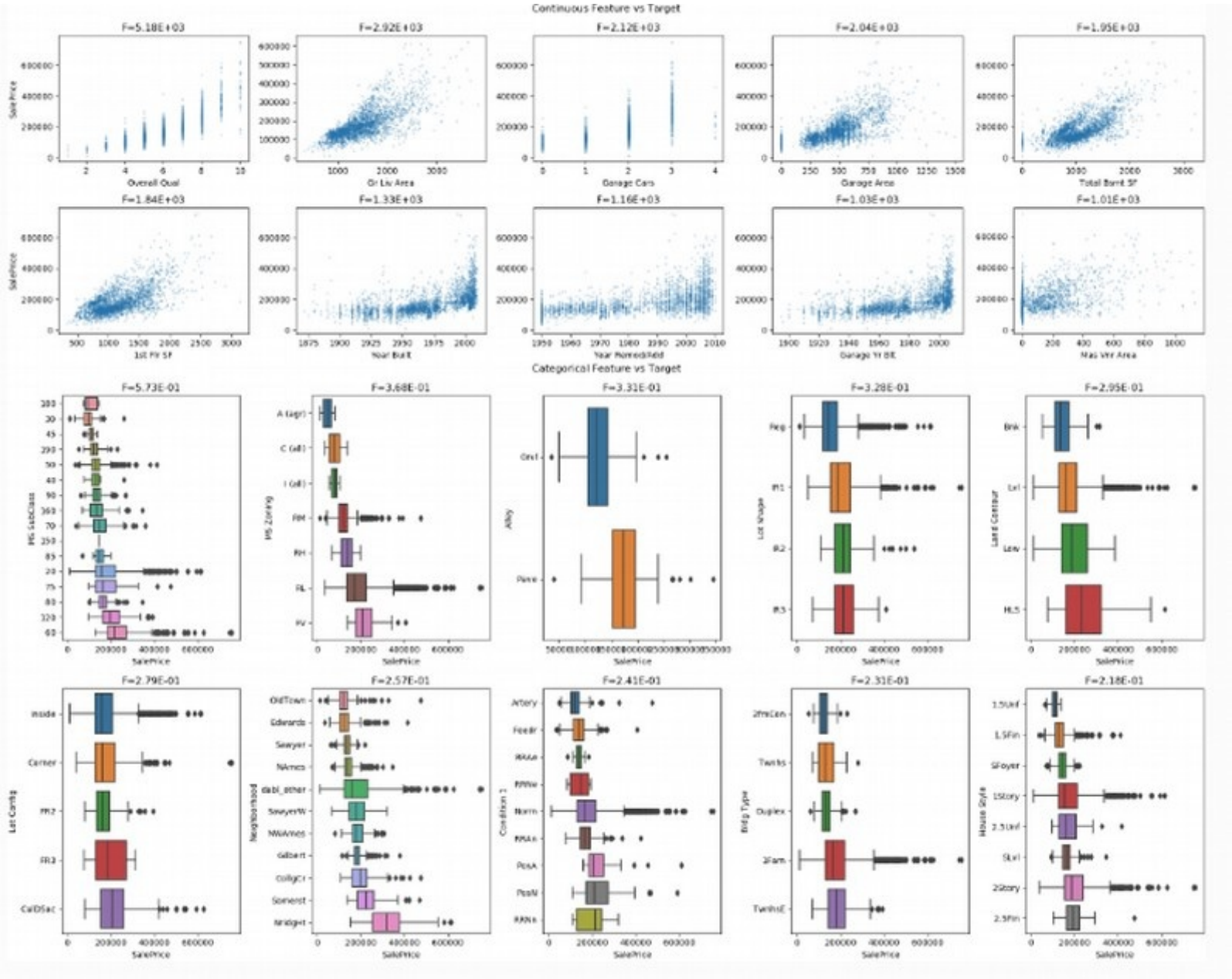
The only thing Keras took from Torch7 is the name of the Sequential class. Checking out early Keras versions demonstrates they have virtually nothing in common. The promo line "in the spirit of Torch" was only meant as "it's simple/minimalist". Sklearn was a far bigger influence.

Current Work

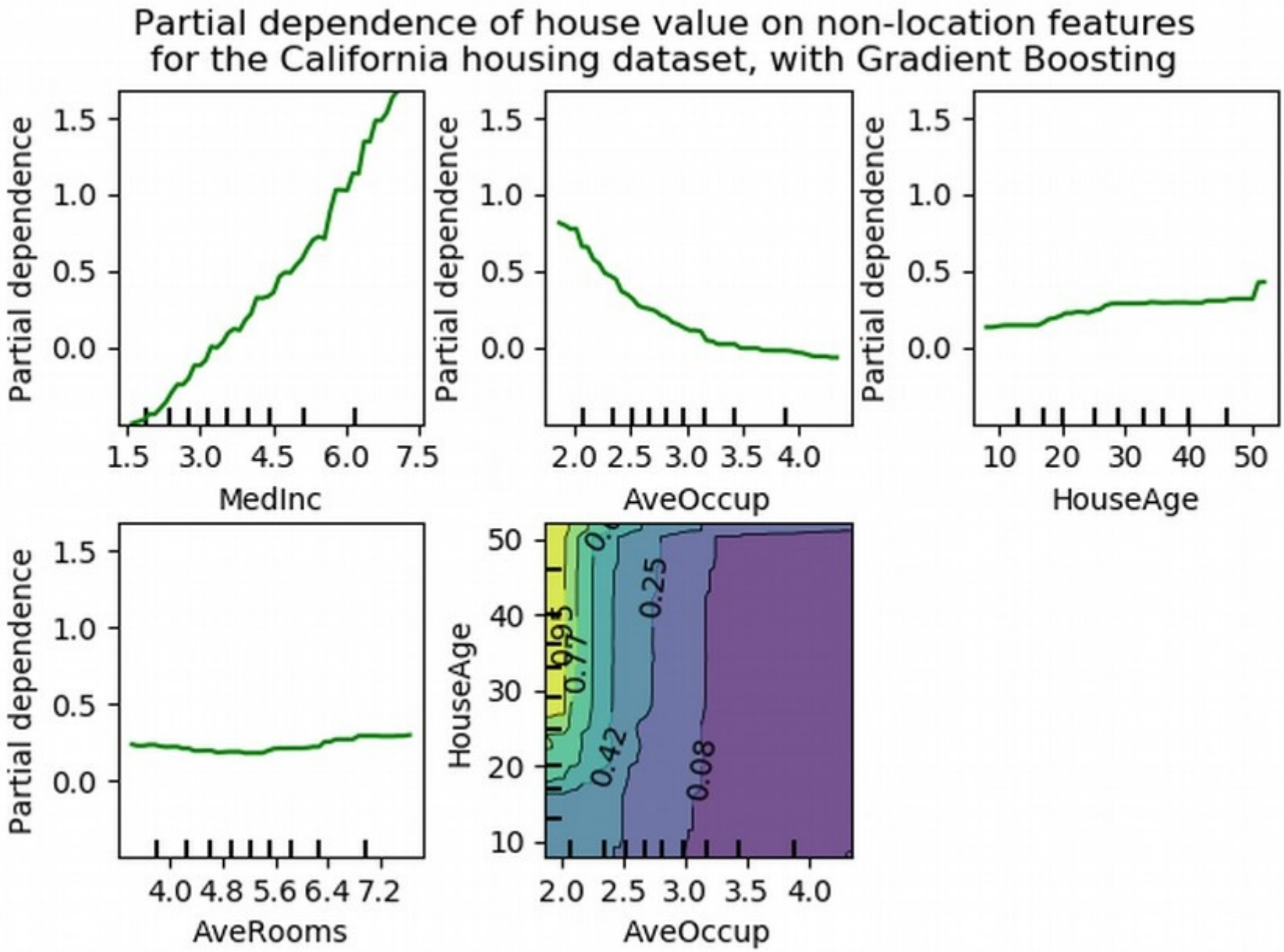
Extreme Gradient Boosting



Visualization



Model understanding and inspection



AutoML & dabl

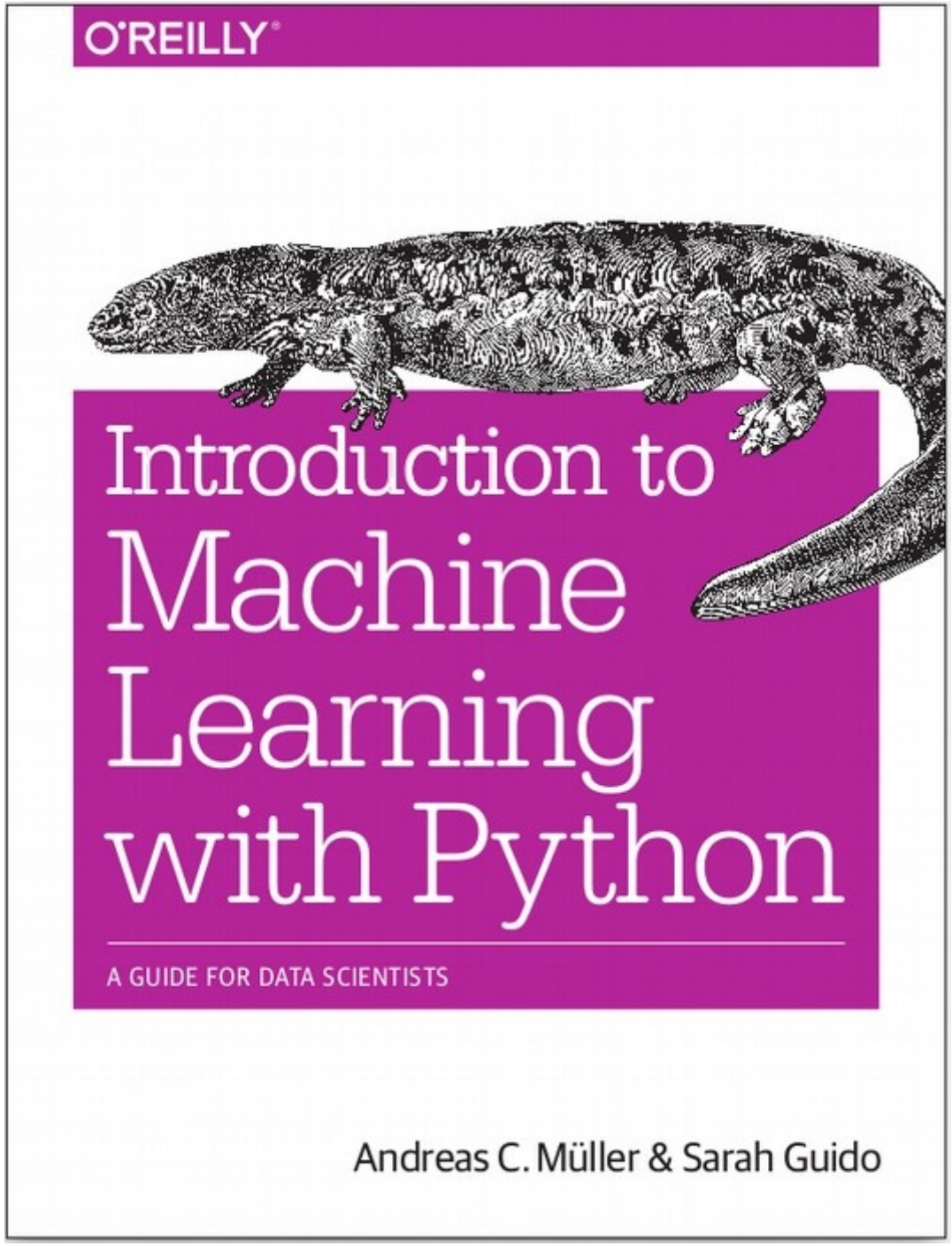
```
>>> types = dabl.detect_types(titanic_clean)
>>> print(types)
                      continuous  dirty_float  ...  free_string  useless
pclass                     False        False  ...        False    False
survived                   False        False  ...        False    False
name                       False        False  ...         True    False
sex                        False        False  ...        False    False
sibsp                      False        False  ...        False    False
parch                      False        False  ...        False    False
ticket                     False        False  ...         True    False
cabin                      False        False  ...         True    False
embarked                   False        False  ...        False    False
boat                       False        False  ...        False    False
home.dest                  False        False  ...         True    False
age_?                      False        False  ...        False    False
age_dabl_continuous         True        False  ...        False    False
fare_?                     False        False  ...        False     True
fare_dabl_continuous        True        False  ...        False    False
body_?                     False        False  ...        False    False
body_dabl_continuous        True        False  ...        False    False

[17 rows x 7 columns]
```

```
sc = SimpleClassifier()
sc.fit(data, target_col=”survived”)

ac = AnyClassifier()
ac.fit(data, target_col="survived")
```

Where to go from here



amueller.github.io

@amuellerml

@amueller

andreas.mueller
@columbia.com